Published In: Physica A 363(2006)1, pp. 104-114

Available online at www.sciencedirect.com
SCIENCE DIRECT®

Physica A 363 (2006) 104–114

PHYSICA A

www.elsevier.com/locate/physa

Autonomous control of production networks using a pheromone approach

D. Armbruster[a,c], C. de Beer[b], M. Freitag[b], T. Jagalski[b], C. Ringhofer[a,*]

[a]*Department of Mathematics, Arizona State University, Tempe, AZ 85287-1804, USA*
[b]*Department of Planning and Control of Production Systems, University of Bremen, D-28359 Bremen, Germany*
[c]*Department of Mechanical Engineering, Eindhoven University of Technology, P.O. Box 513, 5600MB Eindhoven, Netherlands*

Available online 13 February 2006

Abstract

The flow of parts through a production network is usually pre-planned by a central control system. Such central control fails in presence of highly fluctuating demand and/or unforeseen disturbances. To manage such dynamic networks according to low work-in-progress and short throughput times, an autonomous control approach is proposed. Autonomous control means a decentralized routing of the autonomous parts themselves. The parts' decisions base on backward propagated information about the throughput times of finished parts for different routes. So, routes with shorter throughput times attract parts to use this route again. This process can be compared to ants leaving pheromones on their way to communicate with following ants.

The paper focuses on a mathematical description of such autonomously controlled production networks. A fluid model with limited service rates in a general network topology is derived and compared to a discrete-event simulation model. Whereas the discrete-event simulation of production networks is straightforward, the formulation of the addressed scenario in terms of a fluid model is challenging. Here it is shown, how several problems in a fluid model formulation (e.g. discontinuities) can be handled mathematically. Finally, some simulation results for the pheromone-based control with both the discrete-event simulation model and the fluid model are presented for a time-dependent influx.

Keywords: Production networks; Autonomous control; Pheromones; Discrete-event simulation models; Fluid models

1. Introduction

Today, production plans that allocate jobs to machines have to adapt quickly to market dynamics and changing production conditions while conventional production planning and control systems cannot handle these dynamics and unpredictable events and disturbances in a satisfactory manner [1]. One reason is that in practice the complexity of centralized architectures tends to grow rapidly with size, resulting in rapid deterioration of fault tolerance, adaptability and flexibility [2]. To solve this dilemma and to manage the

*Corresponding author.
E-mail addresses: dieter@math.la.asu.edu (D. Armbruster), ber@biba.uni-bremen.de (C. de Beer), fmt@biba.uni-bremen.de (M. Freitag), jag@biba.uni-bremen.de (T. Jagalski), ringhofer@asu.edu (C. Ringhofer).

0378-4371/$ - see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.physa.2006.01.052

dynamics inside and outside the production system, the development of decentralized and autonomous control strategies is a promising research field [3]. Here, autonomous control means a decentralized coordination of intelligent logistic objects (parts, machines, etc.) and the allocation of jobs to machines by the intelligent parts themselves. Those intelligent parts make autonomous decisions that are based on local information. The dynamics of such a system depends on the local decision-making processes and produces a system's global behavior that has new emerging characteristics [4].

The application of autonomous control in production networks leads to a coalescence of material flow and information flow and enables every part or product to manage and control its manufacturing process autonomously [5]. To develop and analyze autonomous control strategies dynamic models are required. For that two different modeling approaches are investigated regarding their abilities to describe an exemplary scenario—an autonomously controlled production network. A discrete-event simulation model is compared to a deterministic fluid model for a continuous product queue, both based on previous work in Refs. [6–9]. Here, the term continuous denotes the continuous material flow in comparison to the flow of discrete parts in the discrete-event simulation model. In literature, continuous flow models of production systems are often called hybrid models (cf. Refs. [10–12]). That means the material flow is modeled as a continuous flow which is controlled by discrete actions. This discrete control is typical for production systems and is applied here in both the continuous and the discrete model.

In this paper, we develop models and control strategies based on the idea of pheromones. That is, the decision which path to choose through the production network is not made by a manager or operator, but by the individual part itself, based on the 'experience' of other parts of the same type. This 'pheromone approach' is therefore similar to, say, the organization of an ant colony, where workers leave chemical messages for the following workers, thereby transmitting the optimal path to the food supply [13]. Clearly, in a very complex network, where the a priori optimal choice becomes an NP-complete problem, this approach offers an interesting alternative.

In Section 2 we encode the topology of a general network with adaptive routing into the language of fluid models. Section 3 is devoted to the description of the discrete-event simulator. In Section 4 we derive the fluid model, which incorporates multiple product flows with separate processing rates for different products. In particular (Section 4.4), we show how to compute the pheromones in the context of a fluid model.

2. The topology

The considered production network is a flow-line manufacturing system producing P different products at the same time. Each of the products has to undergo S production stages. For each of these production stages, we have K parallel production lines available. The raw materials for each product enter the system via sources; the final products leave the system via drains. The production lines are coupled at every stage and every line is able to process every type of product within a certain stage. Thus, at each production stage $s = 1, \ldots, S$, we have to make P decisions, into which of the lines $k = 1, \ldots, K$ to direct product $p = 1, \ldots, P$ in the next stage. Therefore, the production network consists of KS machines. For simplicity, we assume that the service rule for the different products is first in–first out (FIFO). So, each machine has one input buffer in front of it, containing items of P product types. The generalization to the case, where not all products have to go through the same number of production stages, is straightforward, and will not be considered here for reasons of notational simplicity. If all machines at a given production stage would treat each product the same way, then the optimal strategy would be one where each part would choose the line with the currently minimal buffer length. We assume instead, that different product lines are more suitable for certain products. That is, we assume that each machine at each stage has different processing times for each product. Let therefore the processing time for machine (k, s) and product p be given by T_{pks}. To formulate the topology for a continuous model, we have to encode this scenario in terms of the fluid variables, namely the work-in-progress within a buffer–machine system and the fluxes. Let $W_{pks}(t)$ denote the work-in-progress of product of type p in the stage s of production line k at time t, i.e., the number of parts in the buffer plus the number of parts currently processed by the machine. Let $f^{in}_{pks}(t)$ denote the corresponding influx of type p product into the buffer in front of line k at stage s. Finally, let $f^{out}_{pks}(t)$ denote the outflux of type p product out of the (k, s) machine. The

corresponding fluid model is then of the form

$$\frac{\mathrm{d}}{\mathrm{d}t} W_{pks}(t) = f^{in}_{pks}(t) - f^{out}_{pks}(t), \tag{1}$$

expressing the conservation of product. The in- and outfluxes in the network are connected by Kirchhoff's law [14] via

$$f^{in}_{pks} = \sum_{p'k's'} A(p,k,s,p',k',s') f^{out}_{p'k's'} + \lambda_{pks}(t), \tag{2}$$

where the connectivity matrix $A(p,k,s,p',k',s')$ denotes the percentages of the outflux of (p',k',s') going into (p,k,s). Because of mass conservation, assuming a 100% yield for simplicity, we have to have

$$\sum_{pks} A(p,k,s,p',k',s') = 1, \quad p' = 1,\ldots,P, \;\; k' = 1,\ldots,K, \;\; s' = 1,\ldots,S-1, \tag{3}$$

i.e., all product has to go somewhere, except for the last stage, where it just flows out. The term λ_{pks} in (3) denotes the external influx. So λ_{pks} will only be nonzero for $s = 1$ and

$$\sum_{k} \lambda_{pk1}(t) = \phi_p(t)$$

will hold, where ϕ_p denotes the total influx of product p, which is somehow distributed to the lines $k = 1,\ldots,K$ at stage $s = 1$. The product flow is controlled by choosing the connectivity matrix A in (2). We introduce a 'choice variable' $\alpha_{ps}(t)$, which determines which line product p is directed into at stage s. So, we have

$$A(p,k,s,p',k',s') = \begin{cases} 1 & \text{for } k = \alpha_{p's'}(t), \\ 0 & \text{else.} \end{cases} \tag{4}$$

Similarly, the influx vector λ_{pks} is given by the choice at stage $s = 1$:

$$\lambda_{pk1}(t) = \begin{cases} \phi_p(t) & \text{for } k = \alpha_{p1}(t), \\ 0 & \text{else.} \end{cases} \tag{5}$$

This determines the structure of the network. What is left is to model the flux functions $f^{in,out}_{pks}$ and the choice function $\alpha_{ps}(t)$. Note, that, in this setting, all of product p from all lines is directed into line $k = \alpha(p,s)$ in the next stage. The line α_{ps} for product p at stage s will be chosen such that it minimizes the throughput time of previously processed type p parts. How this throughput time is computed in the context of a fluid model is explained in Section 4.4.

3. The discrete-event simulation model

3.1. Description of the discrete-event simulation model

Following the topology as given in Section 2, a discrete-event simulation is set up using the simulation software eM-Plant, where the necessary components, i.e., sources, buffers, machines and drains are predefined. The sources define the entrance of parts to the system following an arrival function that determines the time between the arrival of two parts. The discrete parts queue up in the buffers while they are waiting in front of a machine. They are withdrawn following the queuing policy FIFO. At a machine, the parts are transformed from one status to another which takes a certain processing time T_{pks} depending on the part and machine types. After having passed through all the processing stages the parts leave the system via the drains. As described above the parts have to render a decision at each stage on which flow line they will undertake the next production step. Therefore every machine has to be connected to all the machines on the next stage. The choice function α is modeled as a decision rule every part has to follow to render their decisions. This rule is coded within the simulation tool and runs every time a part leaves a source or a machine.

3.2. Pheromone-based decision rule

The rule is called 'pheromone-based' because it is modeled analogue to the way social insects communicate with the help of pheromones. Comparable to other pheromone concepts (cf. Refs. [5,15,16]), the communication does not take place directly but locally via the environment. Social insects leave an evaporating substance called pheromone on their way and the following insects follow the trail with the strongest pheromone concentration. In contrast to approaches from ant colony optimization (cf. for example Ref. [5]), there is no self-reinforcing guided search process to an optimal solution (e.g. a shortest path in a graph) as in this model the parts have to be able to access updated information about throughput time only. The rule the parts follow in the discrete-event simulation model is to compare the time previous parts had to spend waiting in the buffer plus the processing time on the respective machine. This time will be called total throughput time. So, basically the total throughput time of each part on each buffer–machine system will be stored and made available for parts to render their decision. The decision is then based on the mean total throughput time for the respective part type on the respective machine. To model the evaporation of the pheromone, older data will be replaced. That means that only a certain amount of previous parts will be taken into account for calculation. The number of parts describes the rate of evaporation. This pheromone concept differs from previously proposed concepts for manufacturing control (cf. for instance Ref. [15]), because no reinforcement of the pheromone trail takes place as there is no equivalent to ants returning their way back to the nest. The parts disappear after completing the production steps. Furthermore, this pheromone concept is different from concepts for ant-based routing and load balancing (cf. for example Ref. [16]), as crowding as a consequence of limited capacity is not addressed.

4. The flow model

In this section, we define the continuous model for the machines in the network, i.e., we model the dependence of the outfluxes f^{out}_{pks} in (1) on the influxes f^{in}_{pks} and the work-in-progress W_{pks}. To do so, we use a simple deterministic model for a queue. We assume that each machine has a service rate μ_{ks}.

4.1. The aggregate fluid model

Let $\bar{q}_{ks}$ denote the length of the queue in front of the machine (k,s), containing all the P different product types. Let $\bar{g}_{ks}$ denote the total flux from the queue into the machine. Breaking down the queue length and the fluxes into different product types, we have

$$\bar{q}_{ks} = \sum_{p=1}^{P} q_{pks}, \quad \bar{g}_{ks} = \sum_{p=1}^{P} g_{pks}, \quad k = 1,\ldots,K, \;\; s = 1,\ldots,S,$$

where q_{pks} and g_{pks} denote the portion of the queue and the flux due to the product of type p. We start by defining a deterministic model for the aggregate queue length $\bar{q}_{ks}$. The evolution of the queue $\bar{q}_{ks}$ should switch between the following two types of dynamics:

- If the queue is full ($\bar{q}(ks) \neq 0$) then the evolution should be given by $\bar{q}'_{ks}(t) = \bar{f}^{in}_{ks} - \mu_{ks}$, where $\bar{f}^{in}_{ks} = \sum_{p=1}^{P} f^{in}_{pks}$ denotes the aggregate influx into the machine (k,s).
- If the queue is empty ($\bar{q}_{ks} = 0$), the outflux should be given by $\bar{g}_{ks} = \bar{f}^{in}_{ks}$.

The problem with the above is, that the switch from one type of dynamics to the other has a discontinuous dependence on the queue length $\bar{q}_{ks}$. Therefore, special admissibility conditions would have to be imposed in order to define a unique solution [17]. We avoid this problem by smoothing out the discontinuity in the following way. We define the flux $\bar{g}_{ks}$ from the queue into the machine by

$$\bar{g}_{ks} = \min\left\{\mu_{ks}, \frac{\bar{q}_{ks}}{\varepsilon}\right\},$$

where ε denotes a small smoothing parameter ($\varepsilon \ll 1$), and model the evolution of the aggregate queue by

$$\frac{\mathrm{d}}{\mathrm{d}t}\bar{q}_{ks} = \bar{f}^{in}_{ks} - \bar{g}_{ks} = \bar{f}^{in}_{ks} - \min\left\{\mu_{ks}, \frac{\bar{q}_{ks}}{\varepsilon}\right\}. \tag{6}$$

The evolution equations (6) represent a smoothed out version of the discontinuous model, which replaces zero queue lengths by queue lengths of order $O(\varepsilon)$. This can be seen by considering the following scenario:

- If $\bar{q}_{ks} = O(1)$ (for the full queue) we have, because of $\varepsilon \ll 1$, the equation $(\mathrm{d}/\mathrm{d}t)\bar{q}_{ks} = \bar{f}^{in}_{ks} - \mu_{ks}$.
- As soon as $\bar{q}_{ks}$ becomes small, because the service rate has exceeded the influx for some time, the system switches to $(\mathrm{d}/\mathrm{d}t)\bar{q}_{ks} = \bar{f}^{in}_{ks} - (\bar{q}_{ks}/\varepsilon)$, which, on an $O(1/\varepsilon)$ time scale relaxes against $\bar{q}_{ks}(t) = \varepsilon\bar{f}^{in}_{ks}(t)$ with an outflux $\bar{g}_{ks}(t) = \bar{q}_{ks}(t)/\varepsilon \approx \bar{f}^{in}_{ks}(t)$. This solution only holds of course as long as $\bar{q}_{ks}/\varepsilon < \mu_{ks}$ or $\bar{f}^{in}_{ks} < \mu_{ks}$ holds. As soon as the aggregate influx exceeds μ_{ks}, the system will revert to $(\mathrm{d}/\mathrm{d}t)\bar{q}_{ks} = \bar{f}^{in}_{ks} - \mu_{ks}$.

System (6) can be interpreted as the discretization of a fluid dynamic model for supply chains, presented and analyzed in Ref. [6], by a single Godunov cell, and has been verified extensively against deterministic discrete-event simulations [6].

4.2. The FIFO policy

In order to model the queue lengths and fluxes for the individual product types, we treat the queue as a mixture of different fluids. Thus, we write

$$\bar{q}_{ks} = \sum_{p=1}^{P} q_{pks}, \quad \bar{g}_{ks} = \sum_{p=1}^{P} g_{pks}, \quad k = 1, \ldots, K, \;\; s = 1, \ldots, S.$$

The FIFO policy stipulates that the individual fluid components q_{pks} are processed at the same rate as the whole fluid $\bar{q}_{ks}$. Rewriting (6) as

$$\frac{\mathrm{d}}{\mathrm{d}t}\bar{q}_{ks} = \bar{f}^{in}_{ks} - \bar{g}_{ks}, \quad \bar{g}_{ks} = R_{ks}\bar{q}_{ks}, \quad R_{ks} = \min\left\{\frac{\mu_{ks}}{\bar{q}_{ks}}, \frac{1}{\varepsilon}\right\},$$

where R_{ks} denotes the processing rate, the equations for the different components of the fluid are therefore given by

$$\frac{\mathrm{d}}{\mathrm{d}t}q_{pks} = f^{in}_{pks} - g_{pks}, \quad g_{pks} = R_{ks}q_{pks}, \;\; p = 1, \ldots, P.$$

The outflux of product type p from machine (k, s) is given by a time delay, due to the processing time. Thus we have in (1)

$$f^{out}_{pks}(t) = g_{pks}(t - T_{pks}) = \min\left\{\frac{\mu_{ks}q_{pks}}{\bar{q}_{ks}}, \frac{q_{pks}}{\varepsilon}\right\}|_{t-T_{pks}}.$$

4.3. The service rate

One remaining difficulty is modeling the service rate μ_{ks} of the machines. This problem arises due to the fact that different products will have different processing times, and therefore also different service rates in the same machine. We make the approximation that $1/\mu_{ks}$, the time between service, is a weighted average of the times between service given by the service rates for the individual products. We assume a given service rate μ_{pks} if the buffer (k, s) would only contain products of type p, and compute the average time between service as the average of the 'pure' service times with weighting factors given by the mass fractions $q_{pks}/\bar{q}_{ks}$. So we have

$$\frac{1}{\mu_{ks}(t)} = \sum_{p} \frac{q_{pks}}{\bar{q}_{ks}\mu_{pks}}.$$

So, cf. if there are two products competing for the machine (k,s), with 'pure' service rates $\mu_{1ks}=1,\ \mu_{2ks}=\frac{1}{2}$, and they appear in a ratio of 3 to 1 $q_{1ks}=3,\ q_{2ks}=1,\ \bar{q}_{ks}=4$, the average time between service is given by $1/\mu_{ks}=\frac{3}{4}+\frac{2}{4}=\frac{5}{4}$ and the corresponding service rate is $\mu_{ks}=\frac{4}{5}$. So, the total model for a machine with a FIFO buffer with P products with individual processing times and service rates $T_{pks},\ \mu_{pks},\ p=1,\ldots,P$ is then given by

$$\text{(a) } W_{pks}(t)=q_{pks}(t)+W^{proc}_{pks}(t), \tag{7}$$

$$\text{(b) } \frac{\mathrm{d}}{\mathrm{d}t}q_{pks}=f^{in}_{pks}-g_{pks},\quad g_{pks}=R_{ks}q_{pks},\quad R_{ks}=\min\left\{\frac{1}{\sum_{j=1}^{P}q_{jks}/\mu_{jks}},\frac{1}{\varepsilon}\right\},$$

$$\text{(c) } \frac{\mathrm{d}}{\mathrm{d}t}W^{proc}_{pks}=g_{pks}-f^{out}_{pks},\quad f^{out}_{pks}(t)=g_{pks}(t-T_{pks}).$$

So, the total work-in-progress W_{pks} of the queue-machine system is the sum of the number of parts q_{pks} currently in the queue, and W^{proc}_{pks}, the number of parts currently being processed. Formula (7)(b) has the effect of distributing the total service rate μ_{ks} onto the different product types according to the mass fractions $q_{pks}/\bar{q}_{ks}$. Note, that for a single lot batch processing system, where one part is processed at a given time, and all machines are always running, we have $\mu_{pks}=1/T_{pks}$.

4.4. Throughput time pheromones

To observe the throughput time of parts having just left the machine (k,s), it is necessary to consider the history of the problem. This is done by using so called Newell-curves (N-curves) [18], the antiderivatives of flux functions. We introduce the N-curves

$$\bar{F}^{in}_{ks}(t)=\int_{-\infty}^{t}\bar{f}^{in}_{ks}(r)\,\mathrm{d}r,\quad \bar{G}_{ks}(t)=\int_{-\infty}^{t}\bar{g}_{pks}(r)\,\mathrm{d}r=\int_{-\infty}^{t}R_{ks}(r)\bar{q}_{pks}(r)\,\mathrm{d}r.$$

So, $\bar{F}^{in}_{ks},\bar{G}_{ks}$ is the number of parts having entered/left the queue (k,s) at time t. Since $\bar{F}^{in}_{ks}$ and $\bar{G}_{ks}$ are monotone they have functional inverses, defined by

$$\bar{F}^{in}_{ks}(t)=\sigma\iff t=(\bar{F}^{in}_{ks})^{-1}(\sigma),\quad \bar{G}_{ks}(t)=\sigma\iff t=\bar{G}^{-1}_{ks}(\sigma).$$

Since $\bar{G}_{ks}(t)$ is the (continuous) number of the part leaving the queue at time t, $\bar{G}^{-1}_{ks}(\sigma)$ is the time the part number σ leaves the queue. Therefore, the time part number σ spent in the queue is given by

$$\tau=\bar{G}^{-1}_{ks}(\sigma)-(\bar{F}^{in}_{ks})^{-1}(\sigma).$$

This definition is the same for all part types, because of the FIFO policy. The waiting time in the queue, for a part, leaving the queue at time t is

$$\tau^{wait}_{ks}=\bar{G}^{-1}_{ks}(\bar{G}_{ks}(t))-(\bar{F}^{in}_{ks})^{-1}(\bar{G}_{ks}(t))=t-(\bar{F}^{in}_{ks})^{-1}(\bar{G}_{ks}(t)).$$

The total throughput time through the queue–machine system of a part of type p, which leaves the queue at time t is therefore this expression plus T_{pks}, the processing time for the part. This gives for the total time spent in the queue–machine system by a part leaving the queue at time t

$$\tau_{pks}(t)=\tau^{wait}_{ks}+T_{pks}=t-(\bar{F}^{in}_{ks})^{-1}(\bar{G}_{ks}(t))+T_{pks},$$

and the total throughput time through the queue–machine system of a part of type p, which leaves *the machine* at time t is the above, retarded by T_{pks}, the time it takes to get through the machine, giving

$$\tau_{pks}(t)=t-(\bar{F}^{in}_{ks})^{-1}(\bar{G}_{ks}(t-T_{pks})). \tag{8}$$

The quantity $\tau_{pks}(t)$ constitutes the 'pheromone', the experience transmitted back to following parts. So, the line choice variable in (4) and (5) α_{ps} is chosen to minimize τ_{pks} over all possible lines $k=1,\ldots,K$, and we have

$$\tau_{p,\alpha_{ps},s}=\min\{\tau_{pks},k=1,\ldots,K\},\quad p=1,\ldots,P,\ \ s=1,\ldots,S. \tag{9}$$

5. Simulation results

To demonstrate the pheromone-based control approach, the general network topology is reduced to 2×3 machines, i.e., two production lines each with three stages producing two different products. Considering two parallel lines—each able to process both types of products—gives us on one hand the minimum degree of flexibility necessary to apply the proposed autonomous control. On the other hand, this keeps the parts' decision-making and the resulting buffer levels comprehensible. For special practical application, this configuration can be extended arbitrarily especially by using the discrete-event simulation model.

5.1. Parameter settings

For the 2×3 machines production network, $P = 2,\ K = 2,\ S = 3$ holds. We consider a scenario where the production line $k = 1$ prefers product $p = 1$ and the line $k = 2$ prefers product $p = 2$, i.e., the processing times for product 1 are shorter in line 1 and vice versa. We choose the processing times T_{pks} as

$$T_{11s} = 1, \quad T_{21s} = 2, \quad T_{12s} = 2, \quad T_{22s} = 1, \quad s = 1, 2, 3.$$

We consider a single batch system, so $\mu_{pks} = 1/T_{pks}$ holds. So, both lines together would have a capacity of $\frac{3}{2}$ for each product, if the other product were not present. Conversely, we can reach a steady state when the combined influx of both products does not exceed $\frac{3}{2}$. We choose an influx for product 1 which is above this capacity for a while, as the influx for product 2 oscillates. After a while we keep the influx constant at a level allowing for a steady state. The influx for both products is shown in Fig. 1.

5.2. Implementation

To actually implement the pheromone-based decision rule, outlined in the previous sections, we require a slight modification of the algorithm. Once a preferred line is chosen, and all items are directed into this line, there would be no further updates in the throughput time estimator for the other lines, since no items of the same product type can be observed there. Therefore, the algorithm has to be modified slightly, so as to always direct a small amount of 'test items' into all possible lines to test the throughput time. In the fluid model, this is done by replacing (4) by

$$A(p, k, s, p', k', s') = \begin{cases} 1 - (K-1)\omega & \text{for } k = \alpha_{p's'}(t), \\ \omega & \text{else,} \end{cases}$$

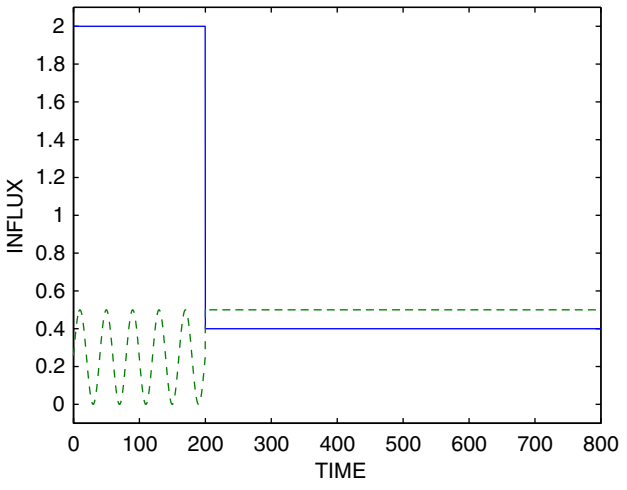

Fig. 1. Influx: solid: product $p = 1$, dashed: product $p = 2$.

for some small parameter ω. Thus, a certain portion ω of the flow is always directed into all possible channels, allowing for a continuous estimation of the throughput time according to Section 4.4. Another issue is the frequency with which the pheromone estimator is evaluated, i.e., how often to observe the pheromone traces. In the discrete-event simulations this is done every five items and in the fluid model this is done in time intervals of $\Delta t = 7.5$—the average processing time of five items. Finally, since it is not sensible to evaluate local in time quantities in a fluid model, which necessarily compute averages, we evaluate estimator (8) for the throughput time as a rolling average over the time window Δt.

5.3. Discrete-event simulation model

The discrete-event simulation model is simulated for $0 \leqslant t \leqslant 1500$ and the chosen lines for each product are observed (cf. Fig. 2). One realizes that the system is slowly reacting. Once a buffer is overloaded the two part types react equally on the rising throughput time because it takes some time until the mean total throughput time of a certain part type is significantly higher than the other. The change of the mean throughput time is not very fast as it is only updated when a part of the same type is finished and the mean is calculated over the last five parts. Therefore, only very few line changes take place. Because of this slowly reacting system the adaptation process to changing situations is not that fast and the parts stay for a longer period on the wrong line. Therefore, the simulation shows a high work-in-progress within the simulation time especially in the first stage and a steady state is not reached within the simulation time. The corresponding work-in-progress of each buffer–machine system at each stage are shown in Fig. 3.

5.4. Fluid model

We simulated the fluid model up to $t = 1500$ and observe convergence to steady state. Steady state in the fluid model is defined by $g_{pks} = f^{in}_{pks}$ in (7)(b), and the products constantly choosing their preferred lines, i.e., $\alpha_{1s}(t) = 1,\ \alpha_{2s}(t) = 2,\ s = 1, 2, 3$ in (4). The choice functions α_{ps} in (4) and (9) for each stage in the buildup phase are shown in Fig. 4. The corresponding work-in-progress of each buffer–machine system at each stage are shown in Fig. 5:

- We observe some moderate amount of switching in the beginning since product 1 learns from the faster or slower throughput times in line 2 depending on how much flux there is of product 2.

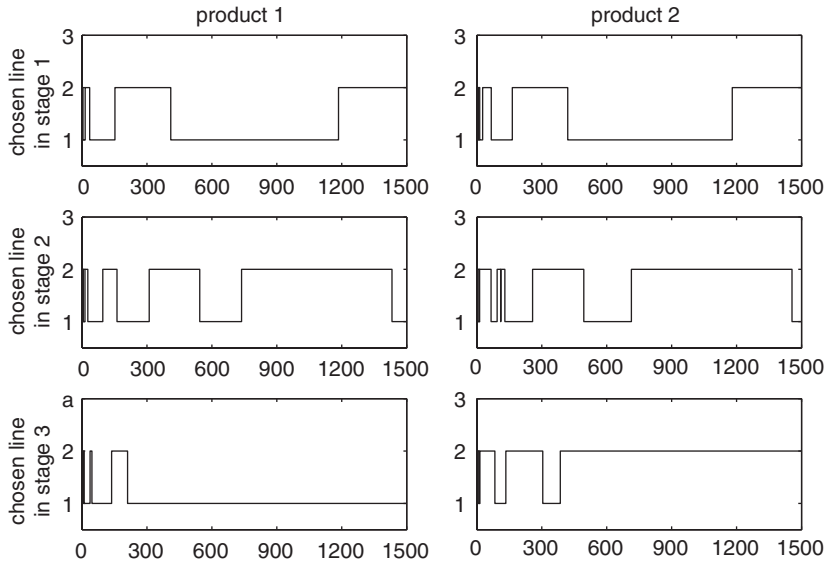

Fig. 2. Line choice $\alpha_{ps}(t)$ using the discrete-event simulation model and the pheromone estimator (9) for the different stages. Left panel: product 1, Right panel: product 2.

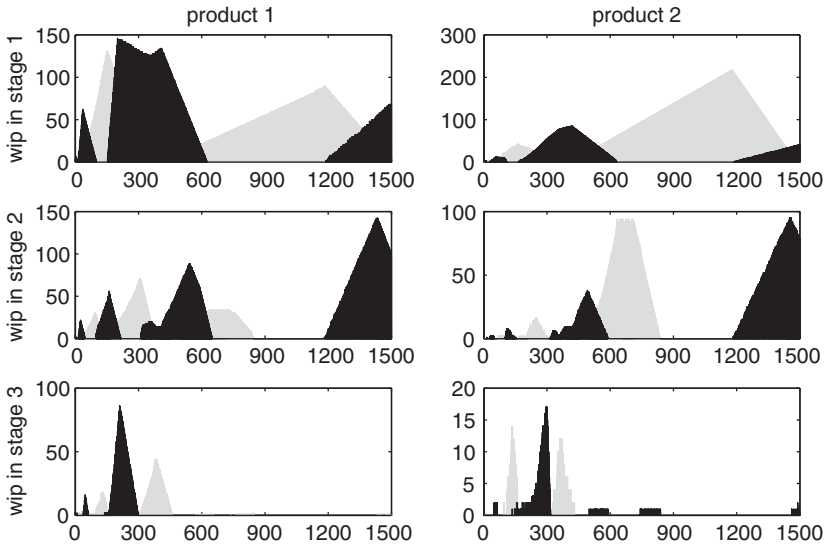

Fig. 3. Work-in-progress (WIP) in the different stages for the discrete-event simulation model and the pheromone estimator. Left panel: product 1, Right panel: product 2. Blue: line $k = 1$. Green: line $k = 2$.

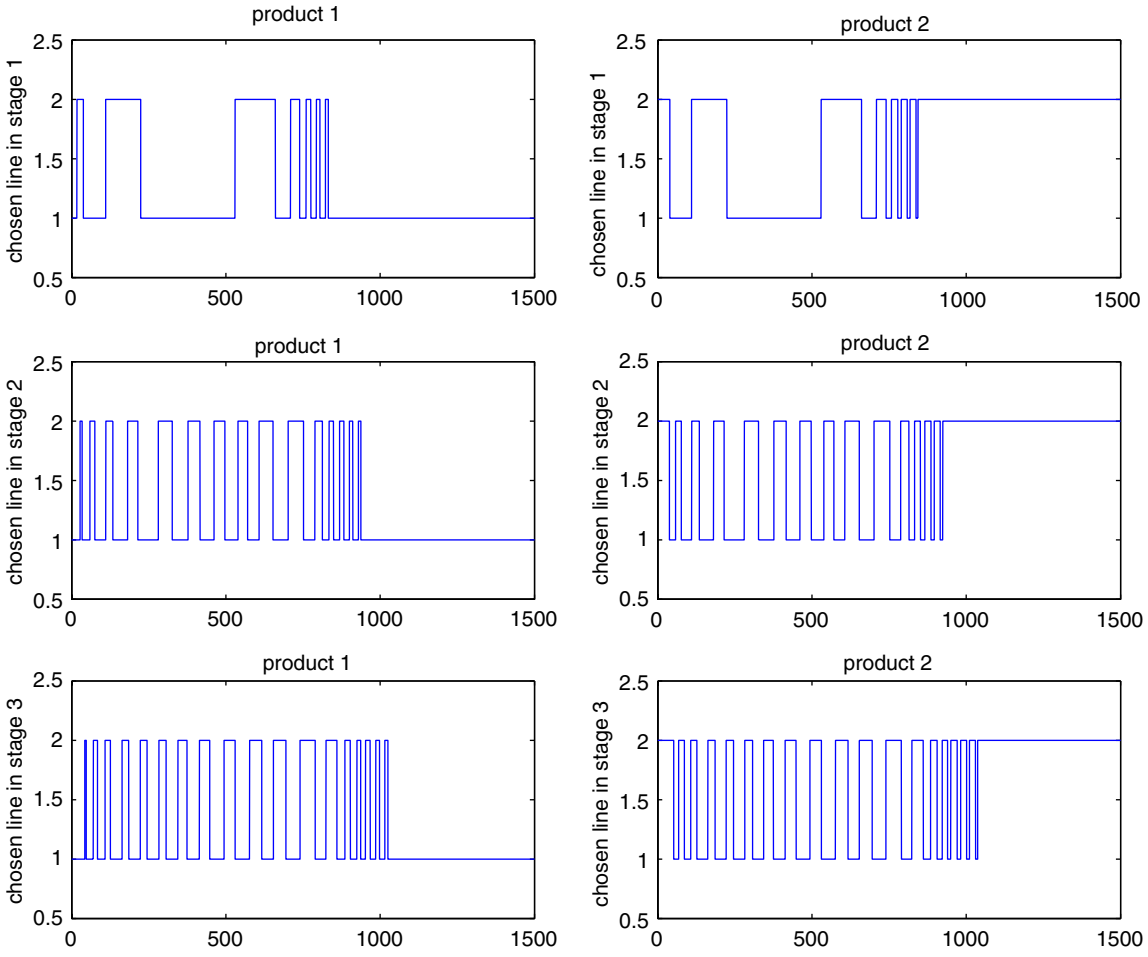

Fig. 4. Line choice $\alpha_{ps}(t)$ using the fluid model and the pheromone estimator (9) for the different stages. Left panel: product 1, Right panel: product 2.

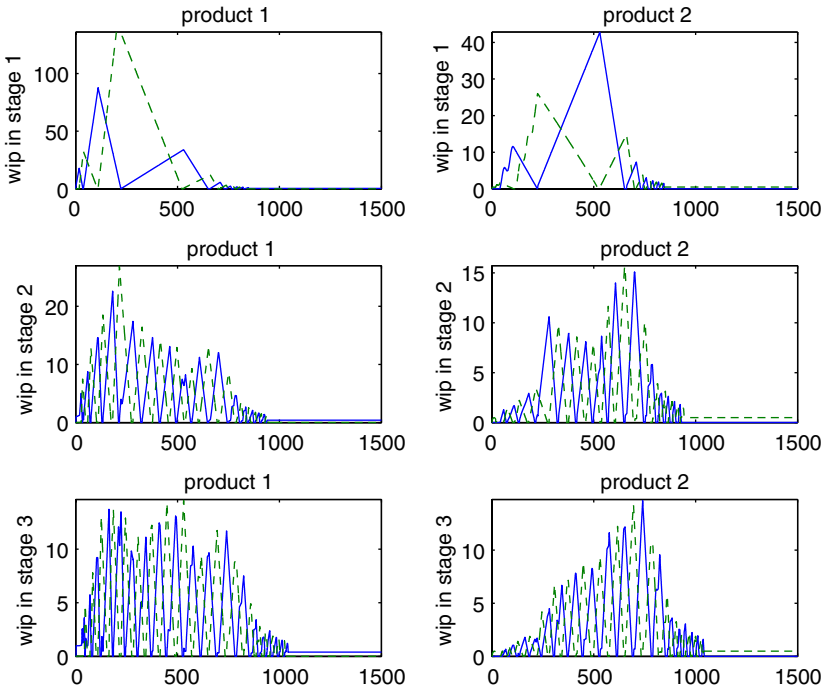

Fig. 5. Work-in-progress (WIP) in the different stages for the fluid model and the pheromone estimator. Left panel: product 1, Right panel: product 2. Solid: line $k = 1$. Dashed: line $k = 2$.

- Roughly at $t = 1000$ the system settles into steady state, with product $p = 1$ constantly choosing its preferred line $k = 1$ and product $p = 2$ choosing line $k = 2$.
- While the fluid model reacts more quickly to changes in the observed throughput time, and therefore the line choices in Fig. 4 are slightly more oscillatory then in Fig. 2, the corresponding work-in-progress in Figs. 5 and 3 agree, more or less, quantitatively. Differences between the discrete-event simulation and fluid model results have to be explained by the approximate choice of the service rate in the fluid model in Section 4.3 and by the averaging of the observed throughput time in Section 5.2.

6. Conclusions

The pheromone-based decision rule is a possible tool for autonomous control of production networks. It is a heuristic learning algorithm based on local (in space and time) evaluation of the throughput time. The paper shows results of the simulation of a fluid model and a discrete-event model of this pheromone-based decision rule. Fluid models require some additional heuristic assumptions in the case that the throughput times of individual machines differ for different product types. This, together with the usual difficulties of comparing discrete and continuous models, such as having to build rolling time averages accounts for some of the quantitative differences.

While they will always have to be verified against discrete-event simulations, one of the advantages of fluid models lies in the fact that they are amenable to an analysis of the underlying dynamics. This will be of major importance in the actual evaluation of the pheromone concept against more classical control strategies. Questions to be addressed, using the groundwork laid in this paper, are:

- The analysis of the stability of steady states and time periodic solutions.
- The influence of random inputs and random service rates (i.e., machine breakdowns).
- A comparison of the efficiency of the pheromone concept (possibly using more complex pheromones) with classical optimal control strategies and model predictive control algorithms.

Acknowledgements

D. Armbruster and C. Ringhofer were supported by National Science Foundation Grant DMS-0204543.

C. de Beer, M. Freitag and T. Jagalski were supported by German Research Foundation (DFG) within the Collaborative Research Centre 637 "Autonomous Cooperating Logistic Processes" (SFB 637) at the University of Bremen, Germany.

References

[1] J.-H. Kim, N.A. Duffie, Backlog control for a closed loop PPC system, Ann. CIRP 53 (2004) 357–360.
[2] V.V. Prabhu, N.A. Duffie, Modelling and analysis of nonlinear dynamics in autonomous heterarchical manufacturing systems control, Ann. CIRP 44 (1995) 425–428.
[3] B. Scholz-Reiter, M. Freitag, K. Windt, Autonomous logistic processes, Proceedings of the 37th CIRP International Seminar on Manufacturing Systems, 2004, pp. 357–362.
[4] D. Helbing, Traffic and related self-driven many particle systems, Rev. Mod. Phys. 73 (2001) 1067–1141.
[5] E. Bonabeau, M. Dorigo, G. Theraulaz, Swarm Intelligence—from Natural to Artificial Systems, Oxford University Press, Oxford, 1999.
[6] D. Armbruster, P. Degond, C. Ringhofer, A model for the dynamics of large queuing networks and supply chains, SIAM J. Appl. Math., 2006, to appear.
[7] S. Dachkovski, F. Wirth, T. Jagalski, Autonomous control of shop floor logistics: analytic models, Proceedings of the IFAC-MIM Conference on Manufacturing, Modelling, Management and Control, 2004.
[8] B. Scholz-Reiter, M. Freitag, C. de Beer, T. Jagalski, Modelling dynamics of autonomous logistic processes: discrete-event versus continuous approaches, Ann. CIRP 55 (2005) 413–416.
[9] B. Scholz-Reiter, M. Freitag, C. de Beer, T. Jagalski, Modelling and analysis of autonomous shop floor control, Proceedings of the 38th CIRP International Seminar on Manufacturing Systems, 2005.
[10] D. Armbruster, D. Marthaler, C. Ringhofer, Kinetic and fluid model hierarchies for supply chains, SIAM J. Mutiscale Model. Simul. 2 (2004) 43–61.
[11] C. Chase, J. Serrano, P. Ramadge, Periodicity and chaos from switched flow systems, Proc. IEEE Trans. Autom. Control (1993) 70–83.
[12] K. Peters, J. Worbs, U. Parlitz, H.-P. Wiendahl, Manufacturing systems with restricted buffer sizes, Nonlinear Dynamics of Production Systems, Wiley, 2004.
[13] H.V. Parunak, Go to the ant: engineering principles from natural multi agent systems, Ann. Oper. Res. 75 (1997) 69–101.
[14] M. Herty, A. Klar, Simplified dynamics and optimization of large scale traffic networks, M3AS 14 (2004) 579.
[15] P. Peeters, H.V. Brussel, P. Valckenaers, J. Wyns, L. Bongaerts, T. Heikkilä, M. Kollingbaum, Pheromone based emergent shop floor control system for flexible flow shops, Proceedings of the International Workshop on Emergent Synthesis, 1999, pp. 173–182.
[16] A. Dussutour, V. Fourcassié, D. Helbing, J.-L. Deneubourg, Optimal traffic organization in ants under crowded conditions, Nature 428 (2004) 70–73.
[17] S. Göttlich, M. Herty, A. Klar, Network models for supply chains, Commun. Math. Sci. 4 (2005) 545–559.
[18] G.F. Newell, A simplified theory of kinematic waves in highway traffic, Transport. Res. B 27 (1993) 281–313.